

S A N A N T O N I O

# SIGGRAPH

2002

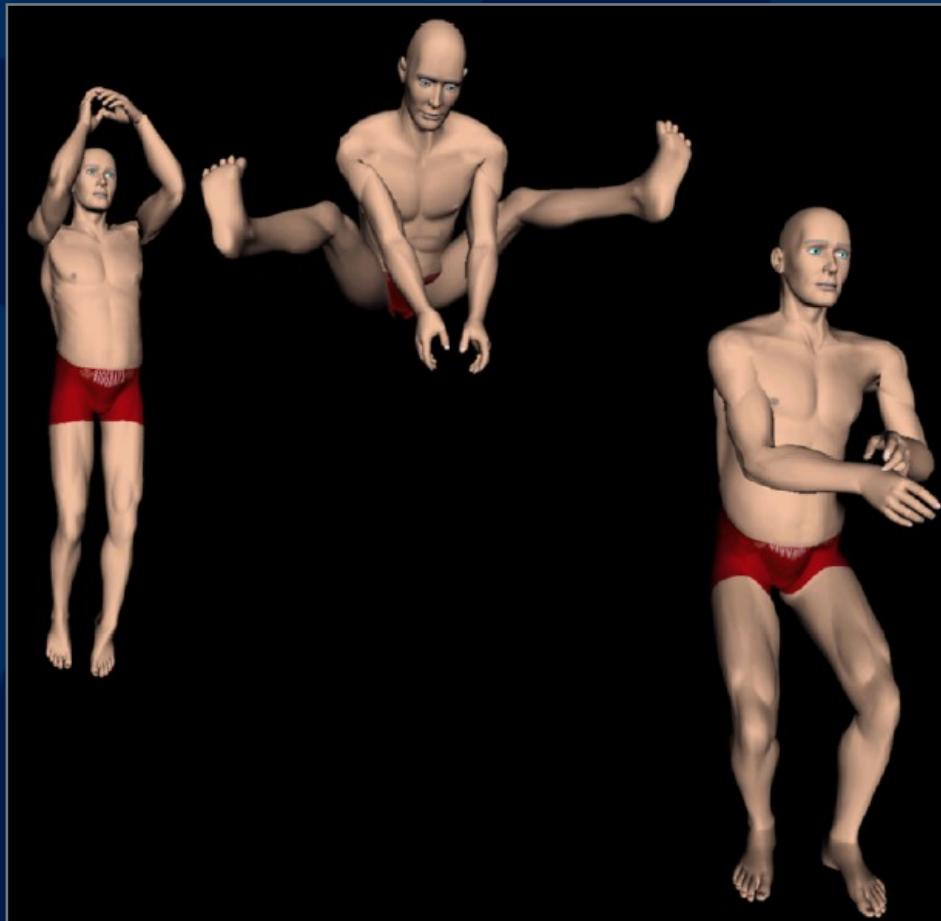
## DyRT

**Dynamic Response Textures  
for Real Time Deformation  
Simulation with Graphics  
Hardware**

**Doug L. James & Dinesh K. Pai  
University of British Columbia**

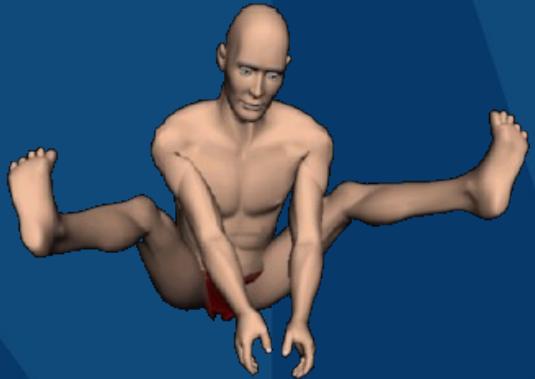
# Motivation for DyRT

S A N A N T O N I O  
**SIGGRAPH**  
2002



# What is DyRT?

- **Dynamic Response Texture (DyRT)**
- **Dynamic physically based deformations**
- **Respond to rigid bone motion**
  - Character animation
- **Synthesized in shader hardware**
- **DyRT is cheap!**

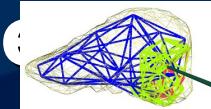


# Related Work

# Related Work: Real Time Deformation

S A N A N T O N I O  
**SIGGRAPH**  
2002

- **Extensive related work** [ Barr84, TerzFleischer88, Chadwick89, Gourret89, PentlandWilliams89, WitkinWelch90, MetaxasTerz92, BaraffWitkin92, WilhelmsVanGelder97, ... ]
- **Multires adaptive approaches** [ZhuangCanny00, Wu01, Debumne01, Picinbono01, ...]
- **Precomputed data-driven deformation**
- **Fast Green's function methods for linear elastostatics** [Cotin99,JamesPai99-02]
  - Multizone [JamesPai ICRA02]
- **EigenSkin, w/ Paul Kry, SCA02**



# Related Work: Modal Simulation

S A N A N T O N I O  
**SIGGRAPH**  
2002

- **Modal analysis is a standard tool**
- **Pentland & Williams,**  
*Good Vibrations, SIGGRAPH 89*
- **Interactive precomputed modal models:**
  - Stochastic dynamics [Stam97]
  - Contact sounds [DoelPai96, DoelKryPai01, O'Brien02]
  - Force-feedback applications [Basdogan01]

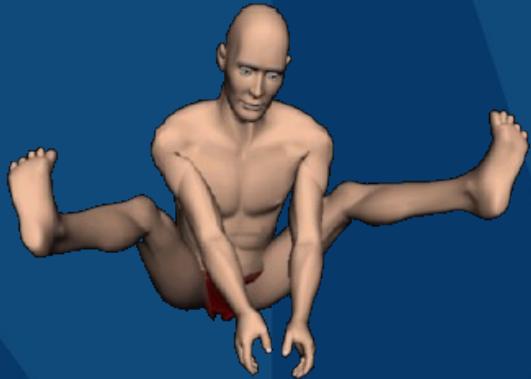
# Our Contributions

- **Dynamic physically based deformations synthesized almost entirely in graphics hardware**
- **Negligible CPU cost**
- **Driven by any rigid motion, e.g., bone-based animation**

# Talk Overview

S A N A N T O N I O  
**SIGGRAPH**  
2002

- **Background: Modal Analysis**
- **Exciting modes with rigid motion**
- **Mapping on to graphics hardware**
- **Recipe for DyRT**
- **Results**



# **Background: Modal Analysis**

## **Interactive Video Tutorial**

# Background: Modal Analysis

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{F}$$

$$\mathbf{u}(t) = \Phi \mathbf{q}(t)$$

$$\left\{ \begin{array}{l} m_i \ddot{\mathbf{q}}_i + c_i \dot{\mathbf{q}}_i + k_i \mathbf{q}_i = 0 \\ \mathbf{u}(t) = \Phi \mathbf{q}(t) \end{array} \right.$$

**Linear elastodynamic  
model**

**Modal transformation**

**decoupled  
DEs**

# Dynamic Integration

S A N A N T O N I O  
**SIGGRAPH**  
2002

- **Simple harmonic oscillator**

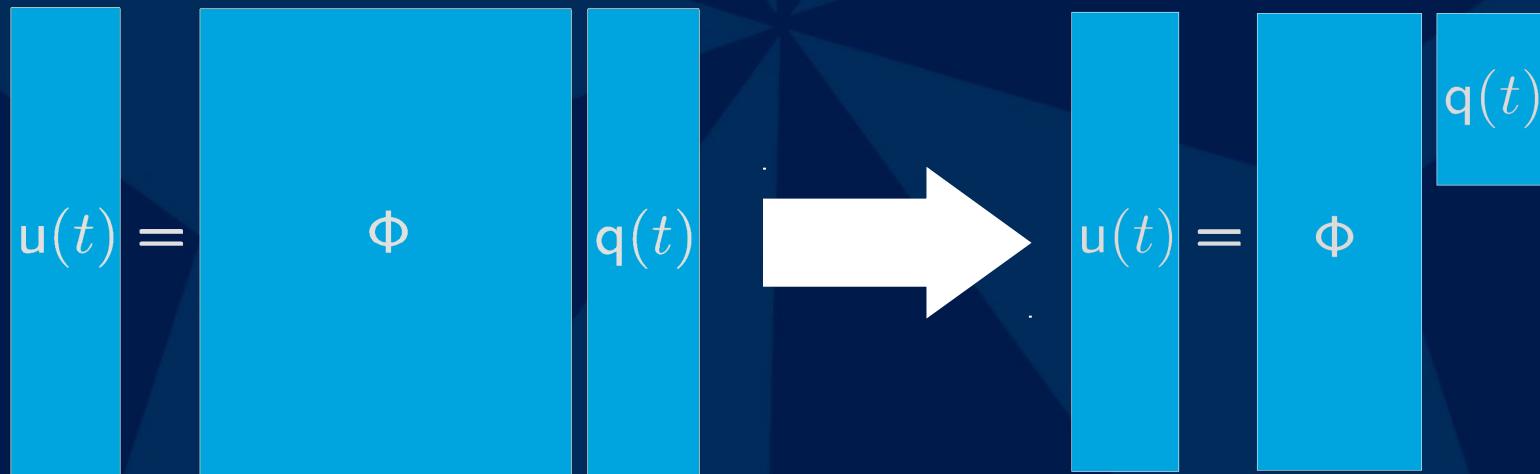
$$m_i \ddot{\mathbf{q}}_i + c_i \dot{\mathbf{q}}_i + k_i \mathbf{q}_i = \mathbf{Q}_i$$

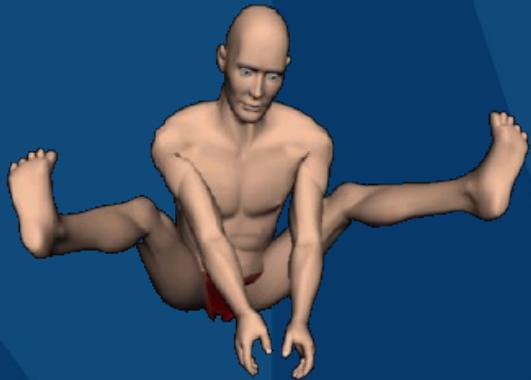
- **Use small IIR digital filter: (mode  $i$ , time  $k$ )**  
$$\mathbf{q}_i^{(k)} = \alpha_i \mathbf{q}_i^{(k-1)} + \beta_i \mathbf{q}_i^{(k-2)} + \gamma_i \mathbf{Q}_i^{(k-1)}$$
- **Precompute IIR filters...**



# Background: Modal Truncation

- **Can often truncate higher modes**
  - Lowest modes are dominant
  - Higher modes heavily damped & die out fast



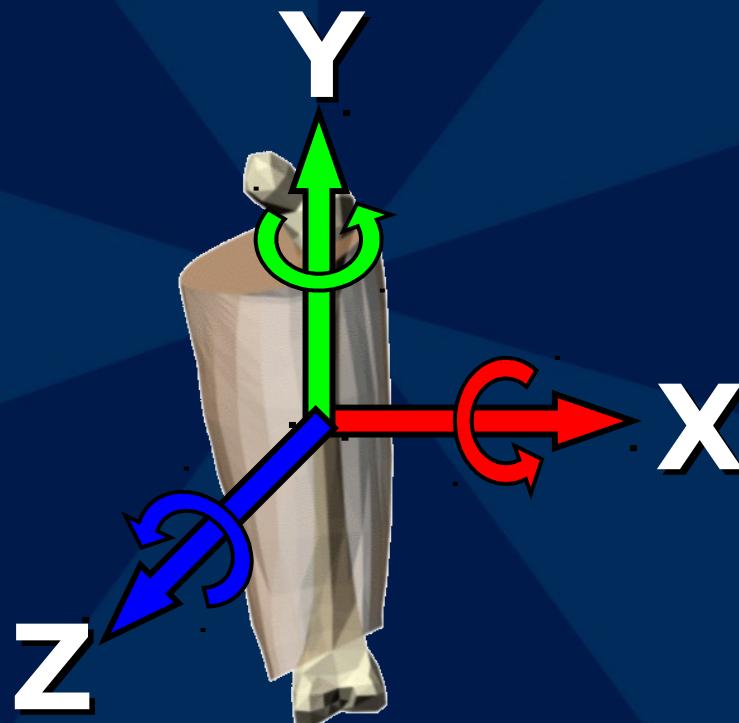


# **Exciting Modes with Rigid Motions**

# Exciting Modes with Rigid Motions

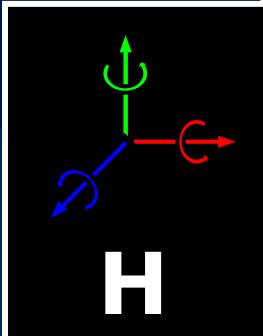
S A N A N T O N I O  
**SIGGRAPH**  
2002

- Accelerations produce inertial forces



# Exciting Modes with Rigid Motions

S A N A N T O N I O  
**SIGGRAPH**  
2002



- **Rigid motion transfer matrix**
  - $m$ -by-6 matrix of precomputed volume integrals

- **Spatial velocity**,  $\psi = \begin{pmatrix} \omega \\ \nu \end{pmatrix}$

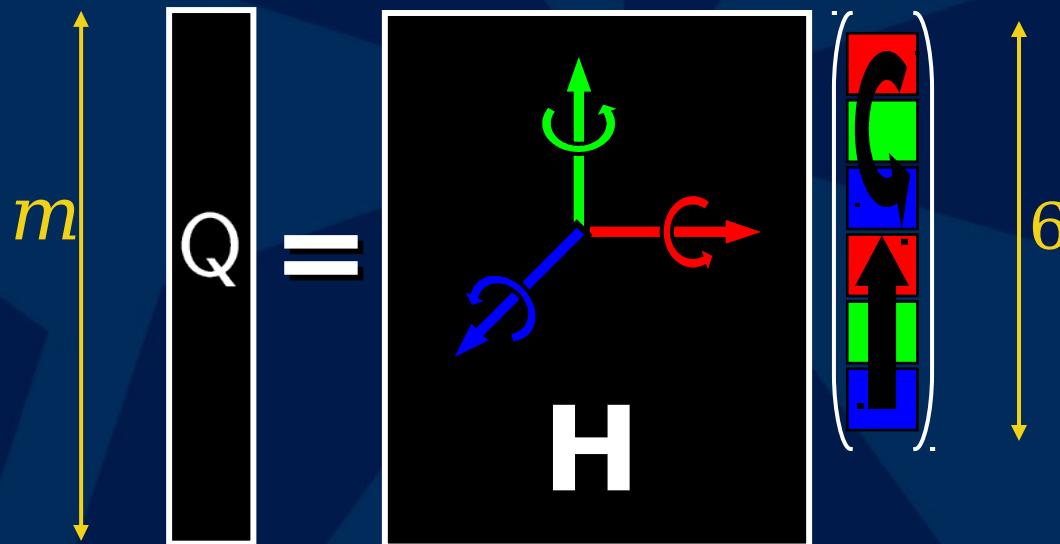
$$(\psi^{(k)} - \psi^{(k-1)}) \xrightarrow{\mathbf{H}} \text{force: } \mathbf{Q}^{(k)}$$

# Exciting Modes with Rigid Motions

S A N A N T O N I O  
**SIGGRAPH**  
2002

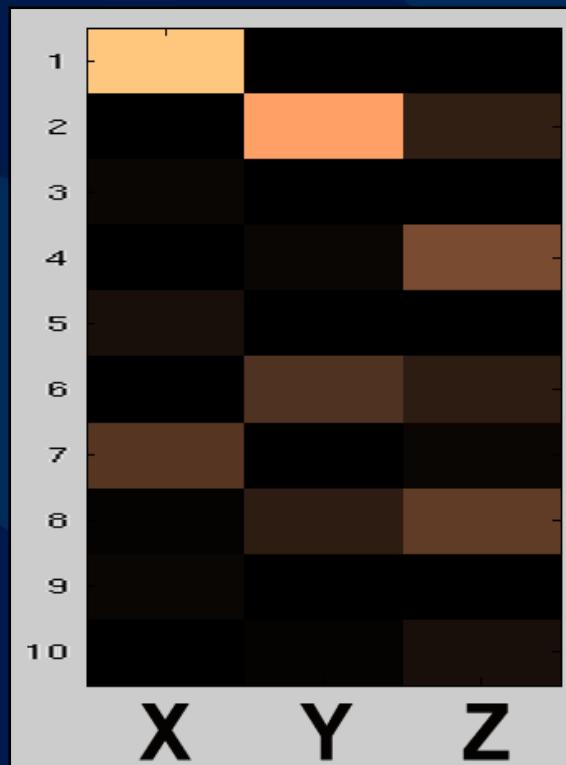
- Matrix picture of modal forces...

$$Q^{(k)} = \mathbf{H}(\psi^{(k)} - \psi^{(k-1)})$$



# Rigid Motion Transfer Matrix, H: Translation

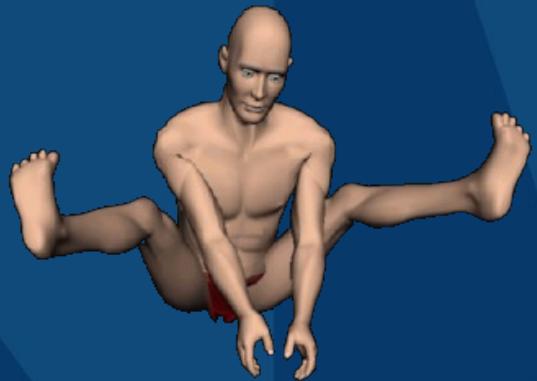
S A N A N T O N I O  
**SIGGRAPH**  
2002



**BELLY**



**THIG**  
**H**



S A N A N T O N I O  
**SIGGRAPH**  
2002

# Mapping on to Graphics Hardware

# Programmable Graphics Hardware

S A N A N T O N I O  
**SIGGRAPH**  
2002

- **Vertex/pixel stream processing**
- **Lindholm, Kilgard & Moreton,  
*A User-Programmable Vertex  
Engine*, SIGGRAPH 2001**
- **DyRT vertex programs**
  - Modes give per-vertex displacements
  - No connectivity available for normals  
⇒ But can easily compute linear correction

# Effect of linearized normal correction...

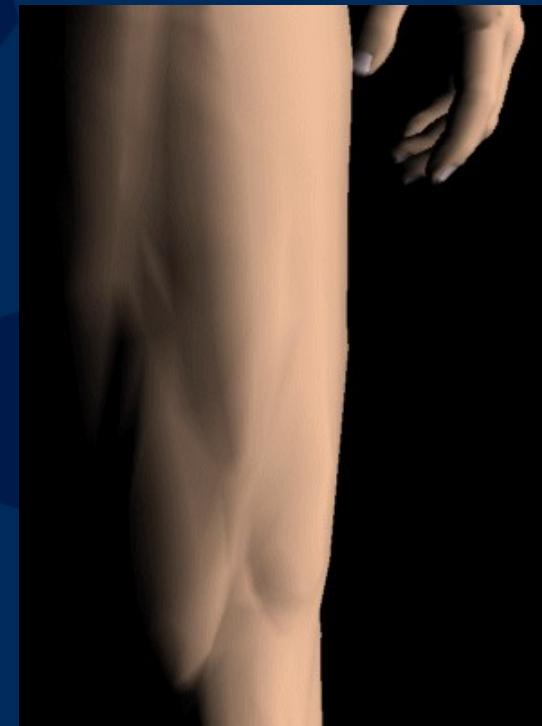
S A N A N T O N I O  
**SIGGRAPH**  
2002



AT REST



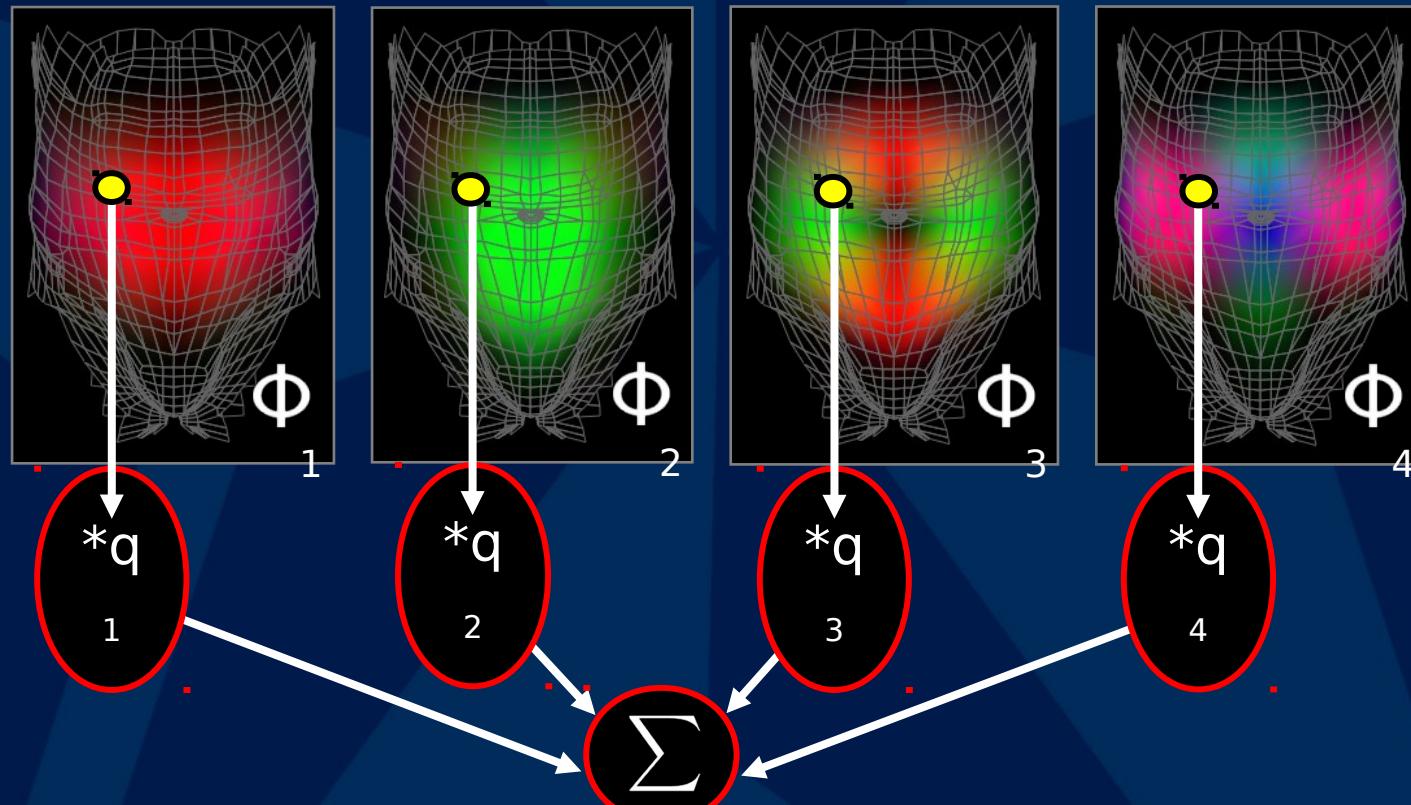
OFF



ON

# DyRT Vertex Program

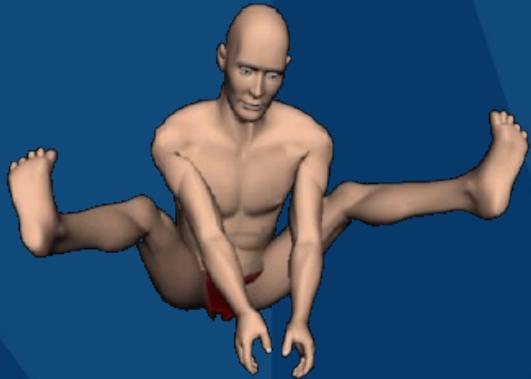
- Per-vertex data  $\Phi_{im}$   $\mathbf{N}_{im}$
- E.g., 4-mode DyRT displacement



# DyRT Vertex Program

S A N A N T O N I O  
**SIGGRAPH**  
2002

```
# Load vertex  $p_i$  into R1 and add 5 modal corrections:  
MOV R1, v[OPOS];                                # R1 =  $p_i$   
MAD R1, c[DyRT ].xxxw, v[5], R1;                # R1 +=  $q_1 \Phi_{i1}$   
MAD R1, c[DyRT ].yyyw, v[6], R1;                # R1 +=  $q_2 \Phi_{i2}$   
MAD R1, c[DyRT ].zzzw, v[7], R1;                # R1 +=  $q_3 \Phi_{i3}$   
MAD R1, c[DyRT+1].xxxw, v[8], R1;               # R1 +=  $q_4 \Phi_{i4}$   
MAD R1, c[DyRT+1].yyyw, v[9], R1;               # R1 +=  $q_5 \Phi_{i5}$   
  
# Load normal  $n_i$  into R2 and add 5 modal corrections:  
MOV R2, v[NRML];                                # R2 =  $n_i$   
MAD R2, c[DyRT ].xxxw, v[10], R2;               # R2 +=  $q_1 N_{i1}$   
MAD R2, c[DyRT ].yyyw, v[11], R2;               # R2 +=  $q_2 N_{i2}$   
MAD R2, c[DyRT ].zzzw, v[12], R2;               # R2 +=  $q_3 N_{i3}$   
MAD R2, c[DyRT+1].xxxw, v[13], R2;               # R2 +=  $q_4 N_{i4}$   
MAD R2, c[DyRT+1].yyyw, v[14], R2;               # R2 +=  $q_5 N_{i5}$ 
```



# Recipe for DyRT

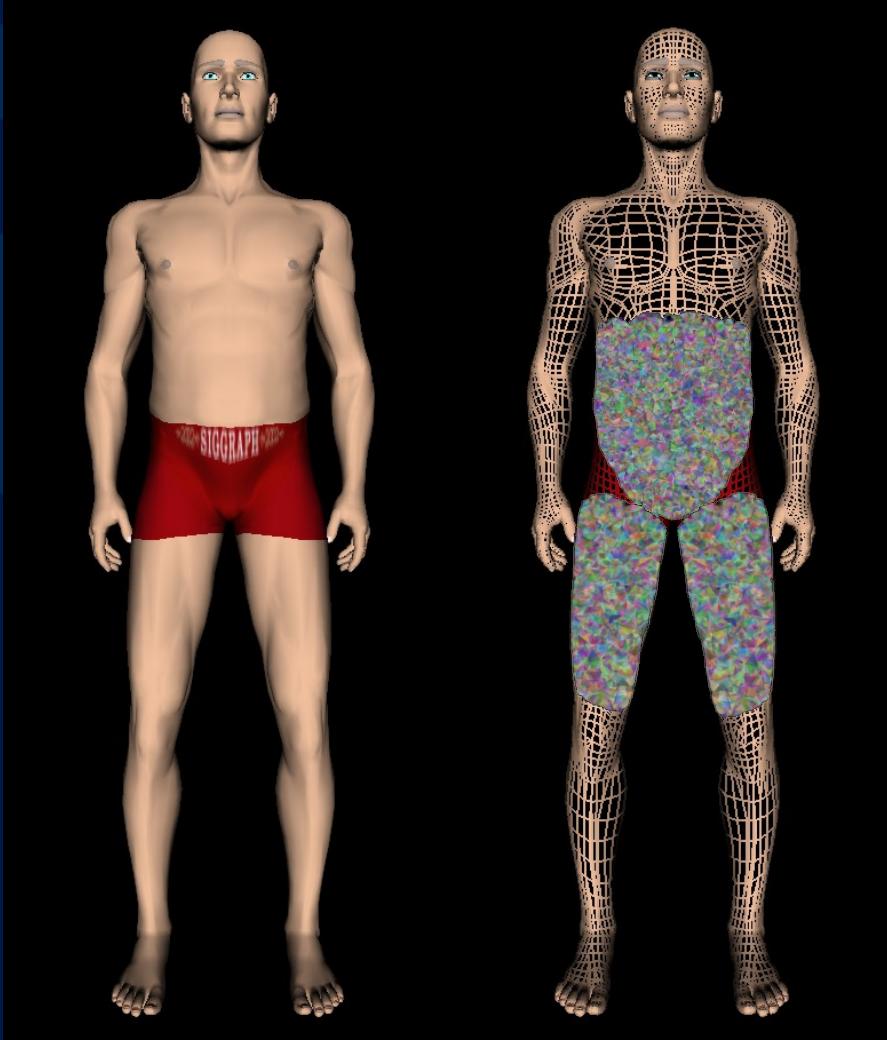
- Precomputation
- Runtime Synthesis

# DyRT Process: Precomputation

- **Define solid model**
  - Anatomically based modeling
- **Finite element modal analysis:**
  - Mesh volume
  - Constrain non-exposed boundary
  - Compute  $m$  leading eigenmodes



# Precomputing DyRT-Man



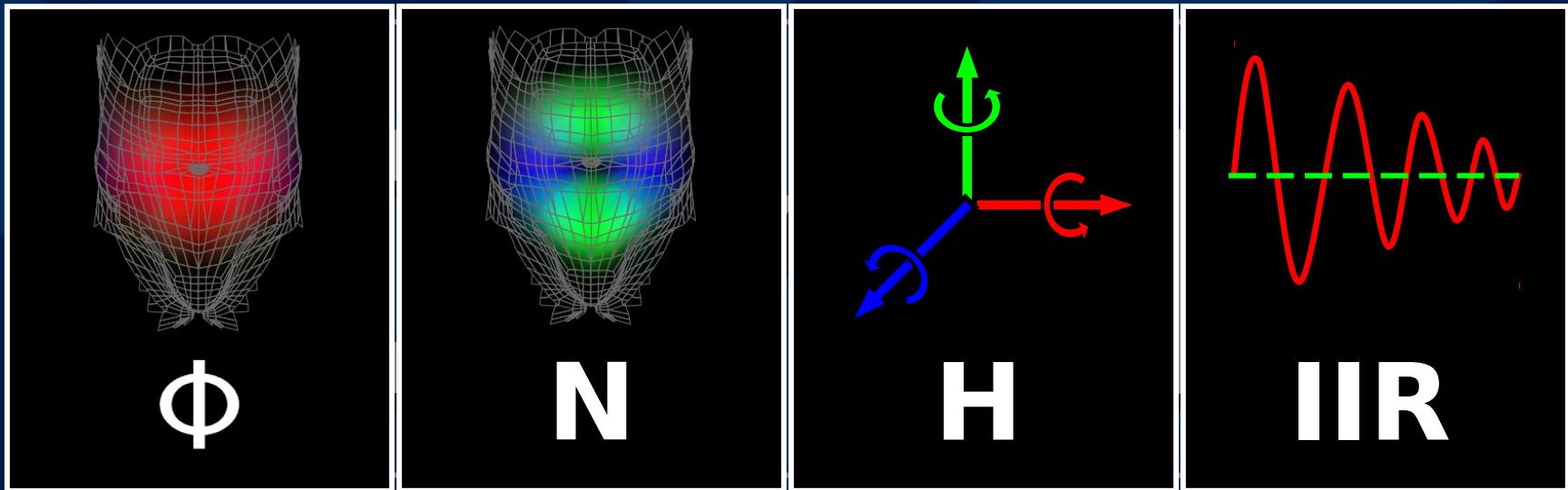
S A N A N T O N I O  
**SIGGRAPH**  
2002

- **Leg model**
  - 10k tetrahedra
- **Belly model**
  - 30k tetrahedra
- **Time: a few minutes**
- **Interpolate modes back onto character surface**
- **Reusable**

# DyRT Process: Precomputation

S A N A N T O N I O  
**SIGGRAPH**  
2002

- Construct  $m$ -mode DyRT object

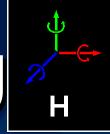


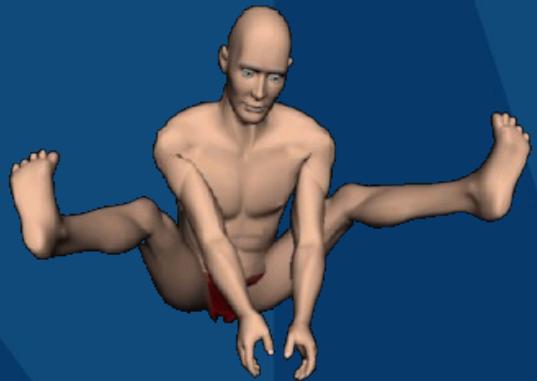
GPU: Per-vertex  
deformation data

CPU: Dynamics

# DyRT Process: Runtime Synthesis

S A N A N T O N I O  
**SIGGRAPH**  
2002

- For each DyRT model...
  - Compute modal forcing 
  - Integrate dynamics   $\Rightarrow q^{(k)}$
  - Bind DyRT vertex program
  - Set program constants ( $q^{(k)}, \dots$ )
  - Draw model, e.g., call display list



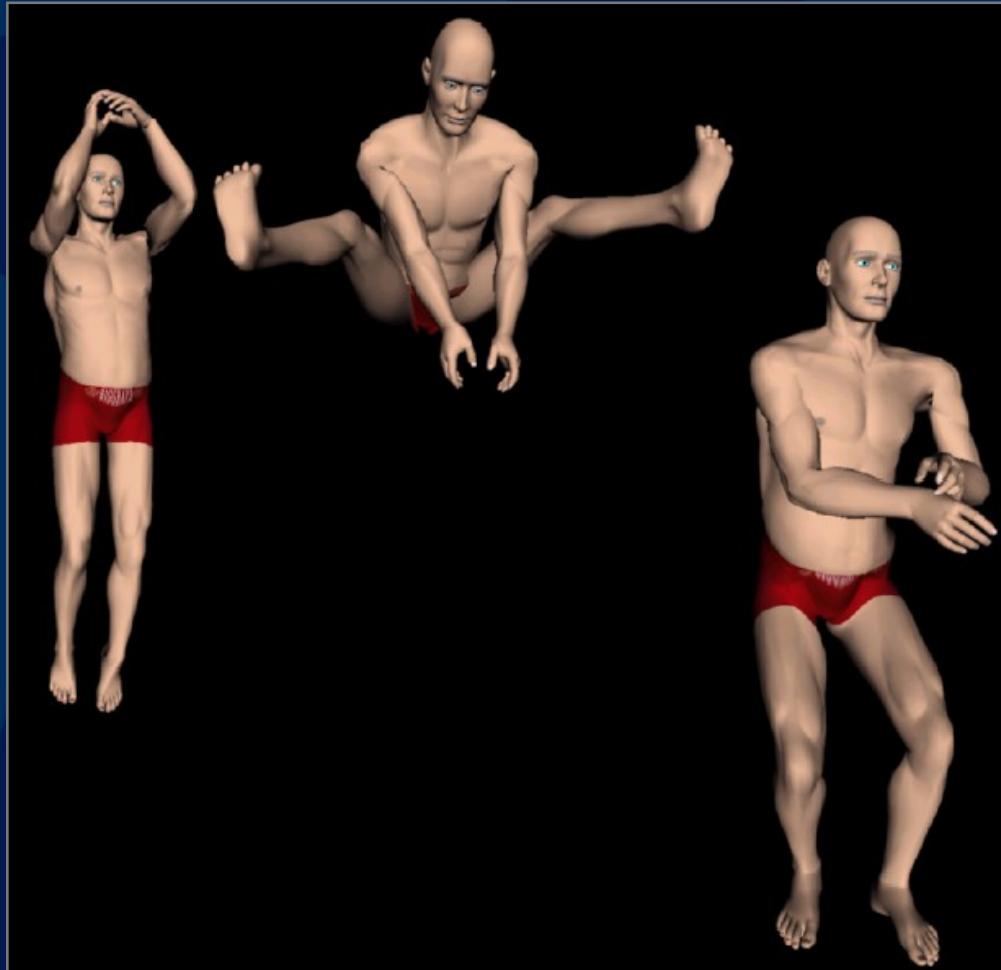
# Results

**Results:** nah-nah nah-nah nah-nah nah-  
nah...

S A N A N T O N I O

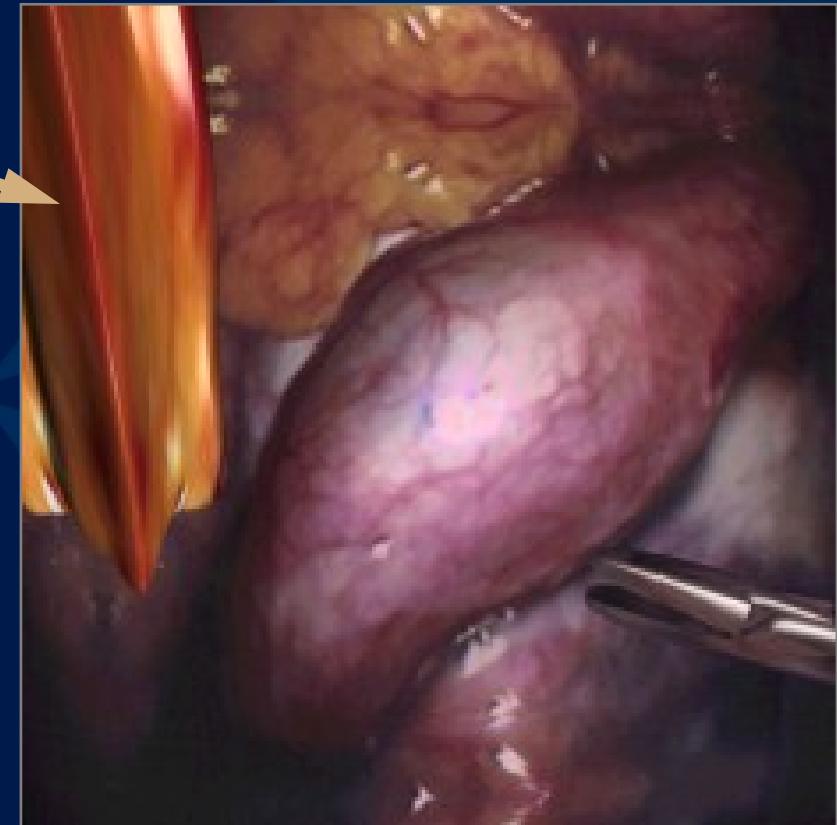
**SIGGRAPH**  
2002

# DyRT-Man!



# Results: Laparoscopic surgical simulation

- DyRT applied to hanging fatty tissue
- Driven by semi-rigid coupling with scene



S A N A N T O N I O  
**SIGGRAPH**  
2002

# Summary & Conclusion

S A N A N T O N I O  
**SIGGRAPH**  
2002

- **Real time dynamic deformations**
- **Responds to rigid motion**
  - Ideal for character animation
- **Synthesized on GPU**
- **Negligible cost to main CPU**
- **Future work...**

**Real DyRT!**

# Acknowledgements

S A N A N T O N I O  
**SIGGRAPH**  
2002

- **Software**
  - House of Moves
  - Curious Labs Poser
  - “GL4Java”
  - FEM software (TETGEN, CalculiX)
- **Funding**
  - Precarn (Inst. Robotics & Intelligent Systems)

S A N A N T O N I O

**SIGGRAPH**  
2002